Information Asset Security Assessment for Software Development and Acquisition

September 6, 2001

DRAFT

Prepared by:
Barre Bull, Technical Director for Information Services
Nikita Black, IT Security Analyst

13921 Parkcenter Rd., Suite 300
Herndon, VA 20171

SAIC-6663-2001-121

Prepared for:
Mr. Greg Montgomery
U.S. Department of Agriculture
Room 431-W
Whitten Building
14th and Independence
Washington, D.C. 20250

For Official Use Only

U.S. Department of Agriculture

Washington, D.C. 20250

USDA Software Development and Acquisition Assessment Guide

1. PURPOSE

This Security Assessment Guide is designed to assist Agency ISSPMs in satisfying their responsibility to develop and implement a comprehensive risk management program as defined in DR 3140-001, "USDA Information Systems Security Policy." By using this guide, Agency ISSPMs can identify areas where Department Information Security requirements are not being met and develop an action plan to ensure all security requirements are satisfied.

2. SCOPE

This guide is to be used by all USDA organizational elements to help assess the security posture of the software development and acquisition environment. The checklist addresses both USDA issued and personal owned systems. This checklist is ***not intended to be a configuration guide*** but a tool to assist in determining if the system meets the requirements for software development and software controls such as internal program controls, operating program controls, and development controls, and assessing the vulnerabilities, both current and potential, they bring to the systems. The checks performed are based on Federal, USDA, and Best Security Practices for the protection of SBU data and systems.

3. BACKGROUND

Risk Assessments are mandated by OMB Circular A-130, Appendix III, "Security of Federal Automated Information Resources." A security risk assessment process is a comprehensive evaluation of the system's technical and non-technical security features. It establishes the extent that a specific design and implementation meets specific security requirements.

4. REFERENCES

a. External
(1) Public Law 100-235, "Computer Security Act of 1987."
(2) Public Law 93-579, "Privacy Act of 1974."
(3) Public Law 93-502, "Freedom of Information Act."
(4) Public Law 99-474, "Computer Fraud and Abuse Act."
(5) OMB Circular No. A-130 Appendix III, "Security of Federal Automated Information Resources," revised February 8, 1996.
(6) OMB Circular No. A-123, "Management Accountability and Control," June 29, 1995.
(7) FIPS No. 140-1 "Security Requirements for Cryptographic Modules," January 11, 1994.

For Official Use Only

b. USDA Internal Regulations
(1) DR 3140-001, “USDA Information Systems Security Policy” dated May 15, 1996.
(2) DR 3300- 1, Appendix I, “USDA Telecommunications and Internet Services and Use,” March 23, 1999.
(3) DM 3140-1 “USDA Management ADP Security Manual” dated March 5, 1992.
(4) DN 3120-1 “USDA Technical Standards Architecture,” dated April 3, 1998.
(5) DR 3200-1, "Application Systems Life Cycle Management"
(6) DR 3200-2, "A Project Manager's Guide to Application System Life Cycle Management"
(7) DR 3200-3, "Software Management."

For Official Use Only

Software Development and Acquisition Assessment Guide

This assessment should be completed by the Agency's ISSPM or designated alternate in conjunction with the Agency Assessment Checklist. Answer all questions. Provide supplemental information as appropriate. All "No" and "Partial" answers must include supplemental information (such as the given reason why the requirement cannot be met) and an action plan that describes how the requirement will be met, as well as a schedule for completion of the plan. Typically, this would be done by developing the action plan in this document and reflecting this in the security plan for the agency.

Agency Identification:

Agency (Agency, Office, Bureau, Service, etc.):		
Address		
Date of last Assessment:		

Test Number: **1**	SITE/SYSTEM:	DATE:	TIME:
Test Name: **Initiation Phase of System Development Life Cycle**			
Resources Required:	Access to system design and interfaces		
Personnel Required:	Management, System Administrators and Software Developers		
Objectives:	To determine if application security has been implemented in the analysis and requirements phase of the development life cycle.		
Procedure Description: (Summary)	Verify complete, unambiguous, and understandable requirements document, the stabilization of requirements as quickly as possible, and the traceability of all requirements from their source to the software requirements document and then through design and implementation and test.		

Detailed Procedures and Results

Step #	Procedure Description	Expected Results	Actual Results (If different from Expected)	Y/N/P
1.	Has a systematic plan been implemented to incorporate life-cycle phases that include an initiation (design) Phase, Development Phase, and Operational Phase?	Guidelines are established to follow a systematic approach of phases for developing new and revising current applications		
2.	Are the following areas considered during the Initiation Phase of the application life cycle? • Objectives • General requirements • Alternative approaches for a target system • Security • Cost-benefit	Expectations that exceed the overall mission requirement will tend to over burden the security requirements making the application reduce it's cost benefit value.		
3.	Have security objectives been classified for the software application?	Major areas of concern for security objectives should include data integrity, data confidentiality, and Automated Data Processing (ADP) availability.		
4.	Has a sensitivity level been assigned to the application to assist with determining security objectives?	Applications should be subcategorized into on e of the following areas and security identified accordingly: 1. General Processing - data integrity		

Step #	Procedure Description	Expected Results	Actual Results (If different from Expected)	Y/N/P
		2. Funds, Accounting, Asset Management - data integrity 3. General-purpose information - integrity and confidentiality 4. Automated Decision Making Systems - rigorous data integrity 5. Real-Time Control Systems - ADP availability and data integrity 6. Systems Affecting National Security or Well-being - ADP support		
5.	Has a systematic review of system vulnerabilities been identified and considered?	The following system vulnerabilities should be addressed early in the initiation phase of the program life cycle. List is not inclusive: - Input errors - Open system access - Poorly defined criteria for authorized access - Unaudited access to data - Unprotected information - Dial-in access - Program errors - Mistaken processing - Operating system flaws - Subverting programs - spoofing		
6.	Have controls been established to achieve application security objectives?	As a minimum, the following areas should be addressed as security control issues: Data validation User identify verification Authorization Journaling Variance detection		

Step #	Procedure Description	Expected Results	Actual Results (If different from Expected)	Y/N/P
		encryption		
7.	Is the security feasibility analysis documented well and clearly? A security feasibility analysis helps to determine if an application can/will successfully meet security needs.	At a minimum, the following areas should be checked to ensure they are addressed during a security feasibility study: • Source data accuracy • User identity verification • Restricted interfaces • Separation of duties • Facility security		
8.	Has the initial risk assessment included determining an estimated loss expectancy compared to the cost and application benefit?	The following areas should be taken into consideration to determine if the risk are worth the cost/use benefits: Impact of major failure - through inaccurate data input - through falsified data - through disclosed data - through lost data - through unavailable data or services		
9.	Has the application security plan included safeguards to consider throughout the development phase?	At a minimum, the following safeguards should be considered throughout the development process: • Application system interface • Responsibilities associated with each interface • Separation of duties • Sensitive objects and operations • Error tolerance • Availability requirements • Requirement for basic controls		
10.	Are data and security requirements identified in accordance with DM3140 and FIPS PUB 73)?	Data and security requirements outlined in DM3140 and FIPS PUB 73 are met.		

11.	Are selections made for appropriate operating systems to host applications?	Operating systems are considered during the early phases of application development.		
12.	Is a review of the security requirements and compliance with security and privacy guidelines conducted and signed off by the Security and Privacy Office during each phase of the System Development Life Cycle (SDLC)?	Security requirements are reviewed throughout the application development process.		
13.	If not present, are there plans to implement formal control procedures in the software programming methodology to ensure all data is reviewed during the quality assurance processes to ensure it is classified and handled appropriately for the level assigned?	Formal control procedures in the software programming methodology have been implemented.		
14.	Are policy documents and security guidelines considered while developing systems?	Security features must be implemented from the beginning.		
15.	Are security requirements included in the demand specification when buying software or developing applications?	The requirements must be included from the beginning.		
16.	Are there procedures for information classification according to the appropriate level of availability? (E.G. open, confidential, secret).	Information classification makes it possible to apply the most effective security measures.		

Comments:
Action Plan:

Test Number: **2**	SITE/SYSTEM:	DATE:	TIME:
Test Name: **Development Phase of System Development Life Cycle**			
Resources Required:	Access to development environment; possible source code and software controls.		
Personnel Required:	Systems Administrator/Software Developer		
Objectives:	To determine is correct and secure software controls have been implemented.		
Procedure Description: (Summary)	Verify that possible error-prone modules have been identified, determine maintainability and reusability of code, and determine adequacy and usability of internal and external documentation.		

Detailed Procedures and Results

Step #	Procedure Description	Expected Results	Actual Results (If different from Expected)	Y/N/P
1.	Are the following stages taken into consideration during the Development Phase? • Design • Programming • Testing • Implementation	Procedures are clearly outlined with timelines to identify the step-by-step process of meeting the three design stages for an applications development.		
2.	Are the following areas considered when addressing application design? • Unnecessary programming • Restricted user interfaces • Human engineering • Shared computer facilities • Isolation of critical code • Backup and recovery procedures • Use of available controls within a program • Design review procedures	All areas are considered. A systematic process of designing an application can help to alleviate many of the unforeseen problems that might arise, thus, reducing the probability of faults in security.		
3.	Are security considerations viewed during each step of the Development Phase?	Security is considered throughout the Development Phase for the purpose of ensuring that appropriate controls and security measures are incorporated in the appropriate locations.		

For Official Use Only

4.	Does documentation identify which features and which settings are open by default?	Programmers should document all features and settings that are open by default during installation.		
5.	Are access points of Inter-process Communications (I-PC creation of temp files throughout a program) methods initialized with proper protection?	Initializing I-PC methods in areas such as socket based communication and shared memory help to prevent other applications from attempting to share memory spaces taken up by the original program. Hackers find vulnerable spots in I-PC breaks and infiltrate their own code into these locations to take control of applications.		
6.	Are considerations taken to avoid the possibilities of buffer overflow?	Precautions are taken to ensure that variable fields for input of data are specifically defined to allow specific data entry or specific character lengths.		
7.	Are potential vulnerabilities documented for use throughout the application programming process?	Identifying possible vulnerabilities throughout an application is required.		
8.	Are authentication and access controls taken into consideration when programming an application?	Users of the system are adequately identified and authenticated throughout the application.		
9.	Are authentication and access controls adjusted to accept user accessibility from other interfaced applications?	In cases where an application is to be used in conjunction with another application, the use of a pre-authenticated user is imperative for certain modules to be efficient and effective. Consideration should be taken to module authentication in areas where specific access is limited.		
10.	Is module level security considered over blanket security?	Modules are designed to accept access control adjustments		

		based upon security needs. For example, a user might not have access rights to the applications configuration section and allowing a blanket access would allow the user to access all areas of the application.		
11.	Are software access points taken into consideration for things such as network interaction, interaction with other applications, and interactions with the operating system?	Access points are established to allow users to interact between programs. Allowing a secure access point into an application allows the user to freely transition from one program to another without having to log on and off the application should the situation prevail where additional authentication is not required.		
12.	Are considerations given to how an application handles temp files for data transfer?	Considerations should be made as to what events occur with data being transferred from an application into a temp file to ensure that the data is deleted following its' use. This will reduce the possibility of classified information from remaining in temporary files.		

13.	Are security philosophies structured towards implementing security features within the software in lieu of relying on the system security features?	Based upon the degree of security required within an application, the application should require an additional authentication process for access. This prevents the casual user from logging into an application that might otherwise be off limits.		
14.	Do applications “zero” out memory when stored information is no longer needed?	Information stored in buffers and registries is zeroed out after a scheduled period of time. This reduces the opportunity for information from remaining in the system registry files when the user has completed using the particular data or application.		
15.	Are interfaces to the system designed to reduce risk exposure?	Interfaces between applications and the user should be designed and developed as to take into account for operator and system error. Developers should attempt to identify if errors might be detrimental to the overall system security or functionality.		
16.	Are code peer reviews documented and used during the developmental stage of an application?	Programmers assigned to application development will request review by one or more peers to ensure that code: • Does not contain errors • Satisfies all design specifications • Is efficient • Is easily maintainable		

17.	Is there documentation on all security-related code?	Documentation is available showing the code that implements security controls, code that performs critical processing, and code that has access to critical or sensitive data during execution.		
18.	Are safeguards in place to ensure that developers do not conduct testing within the production application?	Safeguards, such as access controls, are in place to prevent developers from having access to applications once they are in production.		
19.	Does the developer use software tools (ex. scanning tools) to help identify application security vulnerabilities, application flaws, interface errors, etc.	Programmers select the correct programming languages to support the application development, preprocessors, and debugging.		
20.	Are considerations given to which procedures are used for encrypting/decrypting transit data?	The preferred method would be to encrypt data as it's being saved to the system and decrypting only files being accessed.		
21.	Are "error traps" built into software applications?	Error traps can be used to detect security flaws in code modules, but should be removed once development is complete. Error traps must be documented, and access to them must be strongly controlled.		
22.	Is the programming environment controlled to minimize deliberate errors or deliberate traps, such as "back doors"?	Tests for "back doors" and other deliberate errors/traps should be run against all newly purchased and developed applications.		

23.	Do you document your code thoroughly, including using data dictionaries for full definition of allowable input and output to functions and allowable range and type of values for all variables?	Code is documented throughout the development process, and all data elements are published in the data dictionary.		
24.	Is coding performed uniformly across the product using a defined standard or guideline?	Coding is consistent and uniform.		
25.	Do you sign your source code using digital signatures?	The use of digital signatures prevents unauthorized access to data or source code.		
26.	Do you determine if software is purchased or developed in-house?	Software purchased or developed in-house should contain appropriate security controls to ensure application security.		
27.	If software is developed in-house, do you verify that the software was developed and updated based on the systems development methodology?	All software should be developed and updated based upon SDLC.		
28.	Do you determine how Information Systems controls secure access to the application programs?	System Administrators and developers should be aware of implementation and functionality of application security controls.		
29.	Do you determine what applications interface with the developed application program? Do you document what data is received from and what is sent to these other applications?	System application interfaces and data are checked and verified.		
30.	Do you determine how end-users verify or establish assurances that interfaces are providing complete, accurate and authorized data?	Authorized end-users are encouraged to check and verify that application data is accurate.		
31.	Do you identify and include any mandatory operating system and network security characteristics for the production system in the specifications of the software?	Help guides and users manuals include software specifications and systems requirements.		

32.	Is all software from trusted software application distributors?	All software is from trusted software sources.		
33.	Do you disable all default vendor accounts shipped with the software?	This should be checked after each upgrade or installation.		
34.	Are security controls considered at each step of an application systems life cycle?	Security is considered parallel with system functionality by developers throughout the development of an application.		
35.	Are identified security controls well documented?	Identified security controls are well documented to ensure an understanding of both the intent and outcome of a particular control as it pertains to security of an application.		
36.	Are sensitivity of data and degree of harm that could result from improper action considered as a part of selection of security controls?	When considering implementation of a security control, both sensitivity of data and the degree of harm that could take place should an improper action take place without the control are taken into consideration. When developing security controls, developers take into consideration the extent of damage that might occur should a security control not be implemented.		
37.	Are vulnerabilities considered when selecting security controls?	Both automated system and manual user vulnerabilities are taken into consideration when selecting security controls.		

38.	Are experienced personnel selected to assist with selecting security controls?	Experienced personnel familiar with application interactions with both systems and users are used throughout the application life cycle to help with identifying the most appropriate security controls based upon automated and manual vulnerabilities.		
39.	Have you identified the primary transaction, master and reference files used in processing? Do you work with Security Administrators to determine if these files are secured from unauthorized access?	All files have been identified. Measures to ensure file security is verified with the Security Administrator.		
40.	Do you evaluate how security access restrictions are maintained? Do items of importance include security administration, logon/password management, security monitoring and data ownership?	Security access restrictions are maintained and updated.		
Commercial-Off-The-Shelf (COTS) Products for Software Application Development				
41.	Where applications interact with other applications, has research been conducted to identify "vendor" vulnerabilities?	Developers evaluate vendor concerns and issues pertaining to areas such as memory vulnerabilities, operating system flaws, patches, upgrades, etc.		
42.	Have you determined if any vendor warranties are still in effect?	Vendor warranties are current.		
43.	Is all software from trusted software application distributors?	All software is from trusted software sources.		
44.	Do you disable all default vendor accounts shipped with the software?	This is checked after each upgrade or installation.		
45.	Is there a process in place for evaluation COTS to ensure that it meets organizational security requirements?	Procedures for evaluating COTS software to verify that the software meets security requirements and standards are in		

		place.		
46.	Is there a process in place to ensure the COTS software is pre-tested in a sterile environment to ensure that there are no configuration changes after installing the software?	During development, COTS software is pre-tested for correct functionality and security vulnerabilities.		
47.	Is there a process in place for ensuring that updates and patches are monitored?	All updates and patches are monitored.		
48.	Is there a process in place for ensuring that designated personnel receive the appropriate security patches and upgrades to COTS applications?	Only system administrators are allowed to receive and install COTS software security patches and upgrades.		
49.	Is there a continuous process of evaluating the effectiveness of COTS software to ensure that it continues to operate securely and efficiently?	Procedures to ensure that COTS software is operating effectively and efficiently are in place.		
Database Security for Software Application Development				
50.	Do operating user profiles with sensitive access rights have adequate password controls over it?	The owner of the database and root ID has access rights.		
51.	Are user profiles configured to limit the amount of CPU time, I/O time, connect time, the number of sessions and the amount of memory used by the user, session, or call?	A list of all user definitions from the DBA via the Data Dictionary view DBA_PROFILES is available. All fields are verified for completion.		
52.	Are database audit checks conducted at regular intervals to verify the logical and physical consistency of the database and identify discrepancies such as lost records, open chains and incomplete sets?	Audit checks are conducted and reviewed on a regular basis.		
53.	Are database maintenance utilities that bypass controls restricted and monitored.	Database maintenance utilities are monitored and do not bypass security controls.		

54.	Following an application software failure, is the system capable of automatically recovering the database?	The system is capable of recovering the database in the event of an application software failure.		
55.	Are automated or manual controls implemented to protect against unauthorized disclosure by means of inference search techniques?	Controls, such as suppression and concealing of data, are implemented to prevent statistical inference attacks.		
56.	Is a data dictionary used to document, standardize and control the naming and use of data?	A data dictionary is developed and updated throughout the process.		
57.	Do you ensure that data base applications have been properly reviewed: that the information contained in the data base has been justified by the organization developing the data base, that the required compliance with standards has been met, and that the appropriate levels of data security have been identified.	All databases have been reviewed for purpose of database, verification of accurate data, and compliance of database development standards.		
58.	Do you develop data base systems using a phased approach, with pilot tests to validate design concepts and data element content?	Database development falls within the scope of the SDLC.		
59.	Is the use of ROLES used to assist in controlling access privileges?	A list of all Roles used in the database is available via the view DBA_ROLES_PRIVS.		
60.	Are Database Administrator (DBA) privileges along with the CONNECT and RESOURCE privileges carefully restricted to only those personnel who have responsibility for Database Administration?	The role privileges and the developers in the roles are reviewed for appropriateness.		
61.	Are only DBAs granted access to the SYS DBA role?	Only DBAs have access to the SYS DBA role.		

62.	Do only current developers have access to the database?	Obtain a copy of current employees from Human Resources to verify that database users including developers have not resigned or have not been terminated.		
63.	Are single remote login accounts to the database application used for multiple remote users?	If available, obtain a copy of the database link view, system privileges view, database role privileges view, table privileges view, and column privileges view to verify that none of the roles have the value PUBLIC in the username column. (i.e. Within an Oracle database, the following table views can be checked: DBA_DB_LINKS, DBA_SYS_PRIVS, DBA_ROLE_PRIVS, DBA_TAB_PRIVS, DBA_COL_PRIV) Note: If the value PUBLIC exists verify that there is a valid reason for its use.		
64.	Is object security appropriately set on tables to ensure users only have access to functions need to perform their jobs?	User privileges are appropriately based on their job function based. For example, to check in Oracle, the following table views can be checked: DBA_TAB_GRANTS DBA_COL_GRANTS .		
65.	Are permissions in the database system tables modified?	No users have been granted the SELECT ANY TABLE privilege.		
66.	Is access to the database privileged account restricted? Are application tables created in the “system” tablespace?	Tables are not owned by privileged accounts and applications are not created in “system” tablespace.		
67.	Is use of the WITH GRANT extension limited and used only when a valid purpose is present?	Obtain the appropriate table views and verify that for every row the OWNER column equals		

		the GRANTOR column. For example in Oracle, DBA_TAB_GRANTS and DBA_COL_GRANTS table views can be used to check this option.		
68.	Are the base tables in the Data Dictionary modified?	These tables should never be modified including by the DBA. Obtain a list of base tables contained in the data dictionary and verify that permissions granted on the table do not match privileges granted to users including the DBA.		
Testing for Software Application Development				
69.	Is there a detailed, written test plan, including: - test plan based on requirements - static evaluation -dynamic testing	A detailed test plan for application development is available. Can help to determine if coding is: • Reliable • Meets specifications • Meets the requirements of the user The testing process includes: • A testing plan that identifies what is to be tested, which tools will be used, and what the expected outcomes are. • Static evaluation to analyze the system documentation and code to detect deliberate traps or other unauthorized modifications. • Dynamic testing that includes the		

		execution of the application, or portions of the system to compare the known results to the expected results.		
70.	Are test results of all quality assurance and user acceptance testing reviewed and approved to ensure established test criteria are met prior to implementation of production software?	Test criteria is established and documented.		
71.	When it is not feasible to create test data, are copies of production data used and the confidentiality requirements of the production data adhered to? If production data is used is confidentiality enforced?	Production data may be used as test data if and only if data confidentiality is strictly enforced		
72.	Is "Beta Testing" conducted for newly developed software applications?	Personnel are selected to conduct "Beta Testing" of newly developed applications and revisions to currently used software applications.		
73.	Do you include software testing of all inbound information to ensure exclusion of any data that did not fit the requirements for acceptable data?	This method should be applied to high-risk applications and those with an extremely arduous test cycle and will eliminate many of the common attack methods used.		
74.	Are responsibilities for software quality assurance established?	Software quality assurance should include: • development and distribution of test standards and criteria • performance of quality assurance testing • reporting on test results • custody and retention of test results		

Comments:
Action Plan:

Test Number: **3**	SITE/SYSTEM:	DATE:	TIME:
Test Name: **Operational Phase of System Development Life Cycle**			
Resources Required:	Production application system.		
Personnel Required:	Systems Administrator, Developer, User		
Objectives:	To determine if operating system and software controls are properly configured. To determine if system meets required security standards. To maximize the effectiveness of resources within the project scheduled activities.		
Procedure Description: (Summary)	Verify that the risks determined from the metrics of resource use are based on the appropriateness of the tasks to which resources are being applied at a given time during the life cycle.		

Detailed Procedures and Results

Step #	Procedure Description	Expected Results	Actual Results (If different from Expected)	Y/N/P
1.	Are procedures in place to ensure that application users adhere to security controls included with newly developed software applications?	Users should sign a document to acknowledge that they understand and will adhere to security controls of the software applications.		
2.	Has documentation been maintained current?	Documentation for software development and Commercial-Off-The-Shelf (COTS) products is maintained.		
3.	Has change control been maintained?	Change control is maintained for all application development.		
4.	Are procedures in place to ensure that there is control of data?	Data control in the form of: input verification, data storage management, and output dissemination control are in place.		

Step #	Procedure Description	Expected Results	Actual Results (If different from Expected)	Y/N/P
		• Input verification: Through visual verification, key verification, check digits, control totals, and machine-readable sources. • Data storage: Controlling access to storage areas, authorized users, accounting procedures, backup of sensitive information, and data encryption. • Output dissemination: Controlling output of applications to recipients by logging receipts, from those who receive copies, distribution by mail, and labeling.		
5.	Are contingency plans in place to handle unforeseen down time with an application?	Contingency plans are in place to address what to do when the network supporting an application or system serving an application are down. What course of action to take when the database or application becomes so unusable or unreliable that the data is no longer considered safe or valid.		
6.	Has a security audit activity been established?	A security audit activity has been selected consisting of local personnel experienced in application development. The security audit activity		

Step #	Procedure Description	Expected Results	Actual Results (If different from Expected)	Y/N/P
		will assist developers throughout the application life cycle with identifying initial and progressive changes that should be made to applications based upon changing security needs.		
7.	Has the security audit activity reviewed application controls for reliability in processing data in a timely, accurate, and complete manner?	The security audit activity will review all security controls to ensure that they are handling data in a timely, accurate, and complete manner.		
8.	Has the security audit activity checked to ensure that controls were designed according to specification and legal requirements?	The security audit activity has developed a systematic method for reviewing and documenting checks of security controls to ensure that they meet required specifications and legal requirements.		
9.	Has the security audit activity checked to ensure that controls are operating effectively to provide reliability of and security over data being processed?	The security audit activity has developed a systematic method for reviewing and documenting how reliable security controls are at providing security over the data being processed.		
10.	Are current systems evaluated for potential risks and exposures?	Current applications are continuously evaluated for vulnerabilities, risks, and application flaws.		
11.	Are findings of potential risks and exposures well documented?	Risks and exposures should be documented, updated, and stored as references to prevent future occurrences within the system.		
12.	Is all acquired software examined for viruses, logic bombs or other extraneous malicious features?	Virus scanners for the detection of viruses should scan all software.		

Step #	Procedure Description	Expected Results	Actual Results (If different from Expected)	Y/N/P
13.	Are the same life-cycle phases used for the development of application revisions?	Initiation Phase, Development Phase, and Operational Phase are all a part of the development of application revisions.		
Configuration Management for Software Application Development				
14.	Is your software configuration-management plan (SCM) designed to help programmers and minimize overhead?	A software configuration plan is in place.		
15.	Is application access promptly removed for developers who have left the department?	Once an employee terminates, application access should be removed as soon as possible.		
16.	Is IPSec implemented?	IPSec provides encryption for network sessions using the Internet Protocol (IP) and promises to offer transparent and automatic encryption of network connections.		
17.	Do you use version-control software to facilitate configuration management?	Version-control software protects a program from malicious modification.		
18.	Do you use version-control software to reduce coordination problems of working in teams?	Version control software is used.		
Backups for Software Application Development				
19.	Are software backups taken on a regular basis?	Backups assist in reestablishing lost files, source code and directories in the event of application failure. Backups should be conducted on a daily or weekly basis.		
20.	Are project backups transferred to offsite storage periodically?	Keeping a backup version separate from the system reduces the risk of its loss.		
21.	Are all materials backed up including source code, documents, graphics, and important notes?	All system information should be backed up and safely stored preferably off-site.		

22.	Have you tested the backup-recovery procedure?	The backup recovery procedure		
23.	When using software that access files directly rather than through the raw devices, is the file system remounted as read-only during backups to prevent changes to file access times?	The file system is remounted as read-only.		
Access Controls and Authentication for Software Application Development				
24.	Do you have technical limitations on access to the data, i.e., use on multiple workstations or a network?	Technical limitations on data access are documented.		
25.	Do you have political or legal restrictions on access to the data, i.e., use on multiple workstations or a network?	Political and/or legal restricts on data access are documented.		
26.	Do you verify that every account has a password?	Every account has a password.		
27.	Are test and default passwords changed before user log in?	Test and default passwords are changed before login.		
28.	Are passwords a combination of alphabetic and numeric characters?	Passwords are a combination of alphanumeric characters.		
29.	Do you verify that passwords are at least 6-8 characters long?	Password lengths are verified.		
30.	Are special characters used in passwords?	Passwords are checked for special characters. Some special characters (e.g., # and @) have special meaning to terminal emulation software. Other control characters, like CONTROL-S, CONTROL-H, CONTROL-/, and CONTROL-\e\, can also cause confusion.		
31.	Do you ensure that no two regular users are assigned or share the same account?	To prevent unauthorized access to the system, each user has one account.		
32.	Do you avoid use of the root account for routine activities that can be done under a plain user ID?	Use of the root account for routine activities is avoided.		
33.	Have you established a system by which accounts are always created with a fixed	Accounts are created with a fixed expiration data and must be		

	expiration date and must be renewed in order to be active?	renewed for system access.		
34.	Is there some form of one-time password or token-based authentication, especially on accounts that may be used across a network link?	Accounts needing access across a network link use token-based authentication and/or one-time password.		
35.	Do you enable password constraints, if present in your software, to help prevent users from picking "bad" passwords? Otherwise, consider adding password screening or coaching software to assist your users in picking good passwords.	Passwords constraints are used to guarantee correct password creation. For example, passwords containing names of family and friends, date of birth, etc. should not be used.		
36.	If your computer supports password aging, does the password lifetime conform to policy standards?	It is recommended that the average lifetime of a password should be 90 days or 3 months. For System Administrators and developers, the lifetime should be every month or 30 days.		
37.	Do you disable guest accounts by default and always check to make sure that guest accounts are not enabled?	Guest accounts should be disabled before operation of application. For additional security assign a complex password to the account and restrict its logon access.		
38.	Are passwords, or similar authenticators, obscured by one-way encryption?	Passwords are not obscured by one-way encryption.		
39.	Have you considered removing read access to files that developers do not need to access?	This will prevent unauthorized users from accessing sensitive system files.		
40.	Is there an Identification/Authorization system that controls both users and resources?	There should be a system that controls I&A for users and resources.		
41.	Does the system include access control to resources/objects?	This will prevent the potential modification and fabrication of system resources and objects.		
42.	Are the resource/objects access controls quality tested on password/PIN?	Preferable. Necessary to be able to trace incidents and to get quick alerts.		

43.	Is it possible to reuse old passwords/PIN?	Old passwords should not be reused.		
44.	Is the number of login attempts limited?	The standard limit of login attempts is three.		
45.	Is the system administrator password (root) changed frequently?	The root password is changed every 30 days or every month.		
46.	Does the system block an account if the password is not changed within the time limit or the account has been remained unused?	Users are blocked from system access when passwords are not changed within time limit or unused for an extended period of time.		
47.	Is it possible for a user to change his system privileges?	If the user is not the system administrator, authorization to change system privileges is restricted.		
48.	Is the password/PIN encrypted? (one-way encryption)	The password/PIN has one-way encryption.		
49.	Is the password/PIN individual?	Each user has one password/PIN.		
50.	Do you use the same access password across accounts?	Consider using stronger authentication such as tokens or biometrics for all access to all systems.		
51.	Do you eliminate any duplicate user accounts, test accounts, shared accounts, etc?	Initialize user group policies to assign permissions as needed, and audit accounts regularly.		
Auditing for Software Application Development				
52.	Have you evaluated whether logging on your system is practical and appropriate? If so, install it?	Logging has been proven to be necessary for system security.		
53.	Have you determined if there is an intrusion-detection and/or audit-reduction tool available to use with your logs?	Intrusion-detections and audit-reduction tools have been installed on the system.		
54.	Do you make sure that your log files are on your daily backups before they get reset?	Backups of log files may detect unauthorized access to the system and system resources.		
55.	Do you enable Auditing to alert changes in account policies, attempted password hacks, unauthorized file access, etc.?	Consider auditing the following: *1. Account logon events* *2. Account management* *3. Logon events*		

		4. Object access *5. Policy change* *6. Privilege use* *7. System events*		
56.	Do you make a checklist listing the size, modification time, and permissions of every program on your system?	Include cryptographic checksums in the checklists. Keep copies of this checklist on removable media and use them to determine if any of your system files or programs have been modified.		
57.	If you have backups of critical directories, do you use comparison checking to detect unauthorized modifications?	Protect your backup copies and comparison programs from potential attackers.		
58.	Has software been installed to check message digests of files (e.g., Tripwire)?	File integrity checkers are installed to detect changes in files.		
59.	Are permissions on the security event log set?	Event log files are not protected by default. Permissions should be set on the event log files to allow access to Administrator and System accounts only.		
60.	Is the logging system documented?	The logging system is documented.		
61.	Are the log files protected against unauthorized access?	Access to log files is restricted.		
62.	Is the system configured in a way that the log must be turned on?	Logging is enabled in system configuration.		
63.	What events are logged? -Login -Logout -Failed login -Exceptional behavior -Access Violation -Activities in the Identification and Authorization system -Setting of date and time -Introduction/removal of new hardware -Introduction/removal of files	 -User not acting normally. Might be sorted out via an IDS -Unauthorized access to resources - New users, change of privileges, remove of users etc		

For Official Use Only

Comments:
Action Plan: